

POLITECHNIKA ŁÓDZKA
Wydział Elektrotechniki, Elektroniki,
Informatyki i Automatyki

Praca dyplomowa magisterska

**Analiza problemu komiwojażera przy pomocy algorytmów
heurystycznych**

Traveling salesman problem analysis using heuristic algorithms

Łukasz Tracz

Nr albumu 234011

Opiekun pracy:

Dr inż. Paweł Marciniak,

Dr inż. Cezary Maj

Łódź, 2022

Streszczenie

Abstract

Spis treści

1.	WSTĘP	4
1.1	GENEZA TSP	4
1.2	PROBLEMATYKA ZAGADNIENIA TSP	4
1.3	ZASTOSOWANIA TSP	7
1.3.1	<i>Sekwencjonowanie DNA przez hybrydyzację</i>	7
1.3.2	<i>Optymalizacja ruchów robota do produkcji mikrochipów</i>	8
1.4	TSP A METODY HEURYSTYCZNE	9
2.	CEL PRACY	10
3.	MOTYWACJA	11
4.	ZAŁOŻENIA PRACY	12
5.	PLAN PRACY.....	14
6.	OMÓWIENIE BADANYCH ALGORYTMÓW	15
6.1	ALGORYTM BRUTALNEJ SIŁY.....	15
6.1.1	<i>Naiwne podejście siłowe.....</i>	15
6.1.2	<i>Algorytm Helda-Karpa</i>	15
6.2	ALGORYTM ZACHŁANNY.....	15
6.2.1	<i>Algorytm greedy search.....</i>	15
6.3	ALGORYTM HEURYSTYCZNY	15
6.3.1	<i>Algorytm A*</i>	15
6.3.2	<i>Algorytm przeszukiwań lokalnych</i>	15
6.3.3	<i>Algorytm symulowanego wyżarzania.....</i>	15
6.3.4	<i>Algorytm genetyczny</i>	15
6.3.5	<i>Algorytm mrówkowy</i>	15
6.4	PRZEGŁĄD WYNIKÓW BADAŃ	15
6.4.1	<i>Wyniki badania optymalności.....</i>	15
6.4.2	<i>Wyniki badania złożoności czasowej</i>	15
6.4.3	<i>Wyniki badania złożoności pamięciowej</i>	15
7.	KONKLUZJE	15
8.	BIBLIOGRAFIA.....	16

1. Wstęp

Obecnie transport oraz logistyka to dwa pojęcia bez których świat który znamy dzisiaj w ogóle nie mógł by powstać. Wszystkie państwa na świecie posiadają zawsze jakiś zasób w swoich granicach którego inny kraj w tej chwili wymaga. Taki stan rzeczy pozwolił na rozwój obecnie szeroko pojętego handlu czyli realizacji wymiany dóbr na konkretną ilość danego środka płatniczego. Przedmioty transakcji, zazwyczaj wymagają dostarczenia ich do miejsca przeznaczenia w stosownym czasie i poniesienia jak najmniejszego kosztu związanego z ich transportem.

1.1 Geneza TSP

Problem komiwojażera (ang. Traveling Salesman Problem) to wyzwanie optymalizacji z ograniczeniami tzn. jak znaleźć najlepszy układ zbiorów zmiennych przy znanych regułach i parametrach punktacji. Pomimo że ten problem znany był matematykom od lat to zainteresowanie nim dopiero wzrosło od lat 30 XX wieku. [1] Jednym z pierwszych zainteresowanych tym problemem który ponownie przedstawił to wyzwanie świata był Hassler Whitney z Harvardu, czego dokonał podczas swojego wykładu na Princeton University w 1934r. Po jego wystąpieniu publiczność z USA zapamiętała wyzwanie jako

„48 stanowy problem Hasslera Whitenya”, ponieważ jego wersja problemu zakładała najbardziej optymalnej trasy pozwalającej odwiedzić wszystkie stolice 48 stanów (Alaska i Hawaje wtedy jeszcze nie były stanami). Problem ten był jeszcze badany przez wielu zdolnych matematyków i doczekał się naprawę wiele publikacji naukowych.

1.2 Problematyka zagadnienia TSP

Cała trudność związana z problemem komiwojażera wiąże się z bardzo szybko rosnącą liczbą elementów znajdujących się w naszej przestrzeni rozwiązań wraz ze wzrostem ilości miast, które komiwojażer powinien odwiedzić. Przy założeniu że rozważamy symetryczny problem komiwojażera (droga z punktu A do B jest taka sama jak z B do A) przy grafie posiadającym N wierzchołków, liczba wszystkich możliwych cykli Hamiltona wynosi $(N-1)!/2$. Praktycznie sprawdzenie wszystkich tych kombinacji jest możliwe tylko wtedy gdy graf składa się z niewielkiej liczby wierzchołków. Kolejnym ważnym aspektem jest fakt, że, dopóki cały zbiór wszystkich rozwiązań spełniających nasze

założenia nie został przeszukany, nie jesteśmy w stanie jednoznacznie stwierdzić czy optymalne rozwiązanie już zostało przez nas znalezione i powinniśmy przerwać poszukiwania. Najlepiej wzrost tej trudności opisuje poniższa tabela.

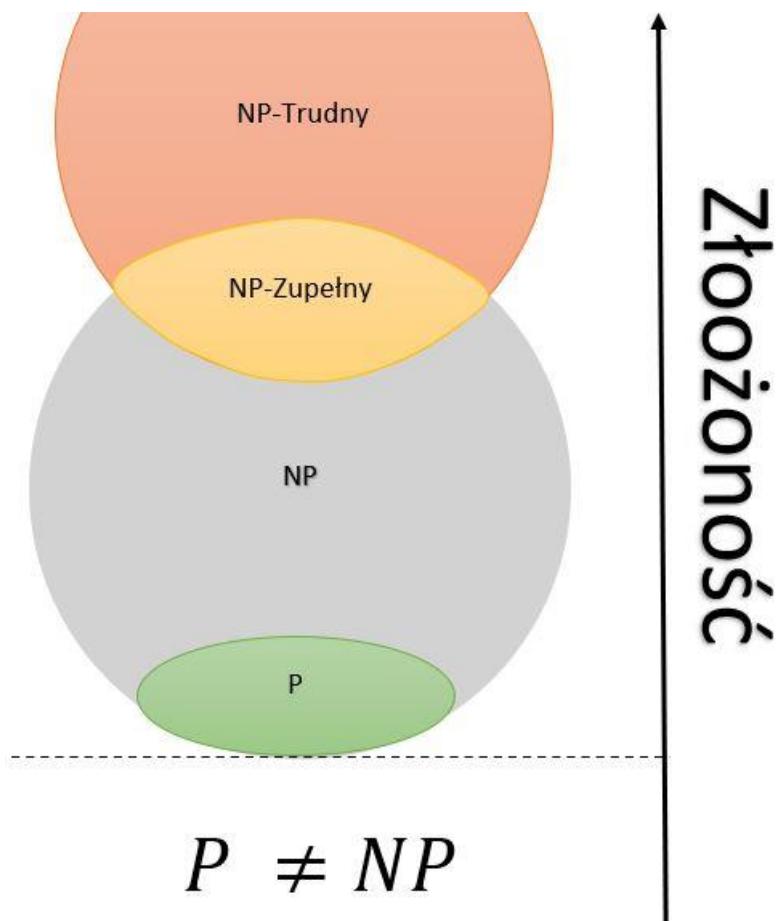
Tabela 1 Tabela przedstawiająca wzrost przestrzeni poszukiwań w zależności od ilości miast dostępnych do odwiedzenia dla komiwojażera

N Ilość miast do odwiedzenia	$\frac{(n - 1)!}{2}$ Ilość wszystkich sekwencji spełniających założenia symetrycznego TSP
4	3
5	12
6	60
7	360
8	2 520
9	20 160
10	181 440
20	$\approx 6 * 10^{16}$

Z uwagi na opisane powyżej trudności, nie istnieje jeszcze ogólny i równocześnie efektywny algorytm zapewniający zawsze optymalne rozwiązanie dla TSP. Wówczas został on zaklasyfikowany do grupy problemów NP-trudnych. Z uwagi na to że istnieje bardzo dużo problemów o złożoności czasowej większej niż wielomianowa, dla problemów NP. utworzono specjalne podzbiory, które umożliwiają dokładniejsze określenie złożoności problemu.

Poniższy rysunek ma za zadanie zwiizualizować oraz wyjaśnić relacje łączące klasę problemu z jego złożonością obliczeniową.

- problem P – problem decyzyjny, dla którego w czasie wielomianowym jesteśmy w stanie znaleźć rozwiązanie
- problem NP – problem decyzyjny, gdzie znamy rozwiązanie problemu, jeżeli w czasie wielomianowym jesteśmy w stanie zweryfikować jego poprawność
- problem NP-zupełny – problem decyzyjny, dla którego niemożliwe jest znalezienie rozwiązania w czasie wielomianowym
- problem NP-trudny – problem obliczeniowy, dla którego nie jest możliwe znalezienie rozwiązania w czasie wielomianowym oraz weryfikacja jego poprawności jest równie trudna co każdego innego problemu z klasy NP.



Rysunek 1 Zależność problemów P , NP , NP -zupełnych i NP -trudnych [10]

1.3 Zastosowania TSP

Pomimo że omawiany problem łatwo odnieść do problemów logistycznych, to jego zastosowanie ma miejsce również w innych dziedzinach, takich jak biologia molekularna oraz produkcja mikrochipów.

1.3.1 Sekwencjonowanie DNA przez hybrydyzację

Przed przystąpieniem do omówienia zastosowania TSP w biologii molekularnej, należy wpierw wyjaśnić trzy terminy DNA, oraz procesy sekwencjonowania i hybrydyzacji.

DNA jest to kwas deoksyrybonukleinowy, jest to organiczny związek chemiczny, który pełni rolę nośnika informacji genetycznej u organizmów żywych i wirusów. Jedną z najważniejszych cech DNA jest jego komplementarność tj. Adenina w DNA zawsze łączy się z Tyminom a Guanina z Cytozyną.

Sekwencjonowanie DNA jest to proces, mający na celu ustalenia kolejności nukleotydów / ciągów nukleotydów wchodzących w skład poddanej sekwencjonowaniu nici DNA.

Hybrydyzacja jest zjawisko spontanicznego łączenia się komplementarnych nici kwasów nukleinowych tj. (DNA z DNA, RNA z RNA lub DNA z RNA). Celem tego procesu jest pod wpływem czynników chemicznych lub wpływu wysokiej temperatury doprowadzić do rozkładu (tzw. denaturacji) pierwotnych nici kwasów nukleinowych do fragmentów a następnie w wyniku usunięcia wpływu czynnika umożliwić ich ponowne łączenie.

Zgodnie z treścią rozprawy doktorskiej dr. Marcina Radom [2], wykorzystując wyżej omówione procesy , chip DNA oraz metody kombinatoryczne, istnieje możliwość odtworzenia kolejności nukleotydów w badanej cząsteczce DNA. W wielkim uproszczeniu proces, ten polega na dokonaniu hybrydyzacji badanej cząsteczki DNA w celu pozyskania

fragmentów DNA. Chip DNA stanowi urządzenie mające za zadanie rozpoznania poszczególnych fragmentów DNA i utworzenia ich cyfrowych reprezentacji. Na tym etapie kończy się faza biochemiczna a zaczyna faza obliczeniowa. Od tego momentu dane uzyskane z chipu DNA stanowią dane wejściowe dla metod kombinatorycznych, których użycie jest niezbędne aby odtworzyć kolejność nukleotydów w poddanej hybrydyzacji cząsteczce DNA. W tym przypadku dla problemu TSP reprezentacje poszczególnych fragmentów DNA stanowią dla komiwojażera miasta, a ilość błędów odzwierciedlenia początkowej nici DNA stanowi koszt uzyskanego rozwiązania.

1.3.2 Optymalizacja ruchów robota do produkcji mikrochipów

Produkcja mikrochipów z uwagi na pracę w małej skali oraz duży stopień skomplikowania układów to bardzo trudny dla człowieka proces. Postępujący rozwój technologiczny sprawił że dzisiaj już mało ludzi ręcznie wytwarza w pełni samemu własne płytki drukowane wlutowując w nie już gotowe układy scalony i elementy elektroniczne by po wykonaniu wszystkich operacji, układ scalony działał zgodnie z wizją autora. Za sprawą postępującej robotyzacji dzisiaj mikrochipy produkuje się przemysłowo na skalę masową. Pomimo, że maszyny nie ulegają zmęczeniu i wykonują operacje wlutowywania elementów taniej, lepiej i szybciej niż człowiek, to nadal można jeszcze dokonać optymalizacji realizowanych przez nich operacji. Jednym z pomysłów by zmniejszyć ilość czasu potrzebnego na wlutowanie dużej ilości elementów na płytke PCB była modyfikacja sekwencji ruchów robota zgodnie z modelem TSP, który to został zbadany w 2003 przez Ratnesh Kumar'a oraz Zhongui Luo na Uniwersytecie w Kentucky [3]. Zgodnie z wynikami ich pracy zastosowanie modelu TSP przy wlutowywaniu elementów na płytke PCB w zależności od liczby elementów i rozmiaru płytki potrafiło skrócić czas wyrobu płytki o około 25% - 33%. Za sprawą rozwiązania problemu TSP ramię robota musiało pokonać najmniejszy możliwy dystans by wykonać wszystkie niezbędne operacje umieszczania elektronicznych komponentów na płytce drukowanej w jak najkrótszym czasie.

1.4 TSP a metody heurystyczne

Po zapoznaniu się z przeszkodami jakie są związane z zagadnieniem TSP, należało by się zastanowić czy ludzkość posiada narzędzia do rozwiązywania tak skomplikowanego problemu. Jednym z tych narzędzi są metody heurystyczne. Aby móc dobrze zrozumieć sposób w jaki rozwiązuje problemy kombinatorycznych, należy najpierw zrozumieć samo pojęcie heurystyki oraz metaheurystyki a także sposobów podziału tych drugich.

Heurystyka (gr. heuriskein – szukam, sztuka znajdowania rozwiązań) powiązane jest z filozofią logiką probabilistyczną, sztuczną inteligencją, teorią informacji oraz metodami przeszukiwania i oceną jakości znalezionych rozwiązań. Jest specjalizowaną metodą rozwiązywania konkretnego problemu, która znajduje dobre rozwiązania przy akceptowalnych nakładach obliczeniowych, ale bez gwarancji osiągnięcia optymalności celu, czy nawet - w wielu przypadkach – jak blisko optymalnego jest otrzymane rozwiązanie.

Metaheurystyka to ogólna metoda służąca za szkielet do konstrukcji heurystyki rozwiązującej dowolny problem, który można opisać za pomocą pewnych definiowanych przez tę metodę pojęć. Metody takie nie służą do rozwiązywania konkretnych problemów, a jedynie podają sposób na utworzenie odpowiedniego algorytmu heurystycznego.

Cechy metaheurystyk [4]:

- opracowywanie strategii określających sposób przeszukiwania przestrzeni dopuszczalnych rozwiązań
- celem działania jest efektywne przeszukiwanie przestrzeni
 - znajdowanie dobrych rozwiązań w określonym regionie (**eksplotacja**)
 - przeglądanie możliwie najszerzego obszaru przestrzeni problemu (**eksploracja**)
- metody przybliżone i zazwyczaj niedeterministyczne
- stosują różne techniki: od prostego przeszukiwania lokalnego do skomplikowanych procesów ewolucyjnych
- wykorzystują mechanizmy zapobiegające utknięciu metody w ograniczonym obszarze przestrzeni problemu
- nie są specjalizowane do żadnego specyficzneego problemu

- wykorzystują wiedzę o problemie i/lub doświadczenie zgromadzone podczas przeszukiwania przestrzeni

Podział metaheurystyk ze względu na [4]:

- Inspiracje
 - Inspirowane przyrodą tj. naśladowujące zjawiska biologiczne i społeczne
 - algorytmy ewolucyjne
 - algorytmy mrówkowe
 - sztuczne systemy immunologiczne
 - optymalizacja rojem cząstek
 - ewolucja kulturalna
 - przeszukiwanie z użyciem zjawiska „tabu”
 - Inspirowane zjawiskami fizycznymi i chemicznymi
 - symulowane wyżarzanie
 - Inspirowane poza przyrodniczo
 - ILS metoda iterowana metodą lokalnych poszukiwań
 - VNS przeszukiwanie zmennego sąsiedztwa
- Ilość znajdowanych przez nie rozwiązań
 - poszukujące wielu rozwiązań (populacyjne)
 - znajdujących pojedyncze rozwiązanie (niepopulacyjne)

2. Cel pracy

Celem pracy było zapoznanie się z istniejącymi już pracami badawczymi opisującymi różne podejścia do rozwiązywania omawianego problemu TSP oraz porównanie ich wydajności oraz optymalności uzyskanych z ich pomocą rozwiązań. Autor do zbadania różnych typów algorytmów zamierza do ich zbadania użyć następujących metryk:

- Czas wykonania algorytmu
- Ilość zaalokowanej pamięci operacyjnej zmierzonych z użyciem narzędzia tracemalloc [5]
- Średnie zużycie procesora do którego pomiaru została wykorzystana biblioteka psutil [6]
- Koszt uzyskanego rozwiązania (suma odległości miast, które odwiedził komiwojażer)
- Błąd bezwzględny (różnica między kosztem uzyskanego rozwiązania a kosztem uzyskania rozwiązania optymalnego (uzyskany minimalny koszt dla optymalnego rozwiązania)
- Błąd względny ([błąd bezwzględny / koszt minimalny] * 100 %)

Autor w swojej pracy zamierza opisać działanie oraz zbadać następujące algorytmy heurystyczne z wykorzystaniem powyżej przedstawionych metryk:

1. Algorytmy brutalnej siły
 - a. naiwny algorytm siłowy
 - b. algorytm Bellmana Helda-Karpa (programowanie dynamiczne)
2. Algorytmy zachłanne (prosta heurystyka)
 - a. Algorytm greedy search
3. Algorytmy heurystyczne
 - a. Algorytm A*
 - b. Algorytm symulowanego wyżarzania
 - c. Algorytm lokalnego przeszukiwania
 - d. Algorytm genetyczny
 - e. Algorytm mrówkowy

3. Motywacja

Główny motywem wyboru owego tematu przez autora było uzyskanie możliwości poznania już istniejących sposobów rozwiązywania problemu komiwojażera w oparciu o różne dokonania informatyczne i matematyczne poczynając od wykorzystania podstawowej wiedzy z teorii grafów i kombinatoryki aż po wykorzystanie programowania dynamicznego, algorytmów heurystycznych w tym algorytmów ewolucyjnych.

Autora również cechuje zamiłowanie do tworzenia oprogramowania i chęć ciągłego rozwoju dlatego jako język programowania do realizacji swoich badań wybrał on język Python, który pozwala na programowanie funkcjonalne jak i obiektowe oraz posiada dostęp do szerokiego zakresu gotowych bibliotek pozwalających na wykorzystanie i porównanie gotowych już zaimplementowanych algorytmów heurystycznych i opracowanie do ich porównywania stosownej programistycznej architektury pomiarowej.

Pomimo, że przy rozwiązywaniu problemów optymalizacyjnych powinien być wybrany język, który jest uważany za najszybszy (obecnie C/C++), to autor zdecydował się wykorzystać język Python, z uwagi na jego wciąż rosnącą popularność oraz tym że w roku 2021 zgodnie z raportem „The software quality company” [9] wyprzedził on język C

i C++ jak również zostawił on uwielbianą przez szerzę fanów tworzących aplikacje Enterprise Javę w tyle.

4. Założenia pracy

W celu zrealizowanie postawionych celów należało zapewnić powtarzalność otrzymywanych wyników. do tego celu autor postanowił utworzyć pliki z danymi wejściowymi utworzone zgodnie z poniższymi założeniami.

Założenia dotyczące plików z danymi wejściowymi:

- każdy plik wejściowy zawiera dane wejściowe w formacie JSON
- do badań zostanie wykorzystanych po 100 plików z danymi wejściowymi na każdą ilość badanych miast
- do badania zostaną wykorzystane pliki zawierające od 4 do 15 miast (wartości x i y znajdujące się w układzie kartezjańskim 2000 x 2000)
- każdy plik zawiera miasto znajdujące się w początku układu współrzędnych
- każdy z plików będzie dodatkowo zawierał następujące informacje statystyczne na temat położenia miast:
 - maksymalna wartość x oraz y
 - minimalna wartość x oraz y
 - odchylenie standardowe dla x oraz y
 - wartość pierwszego kwartyłu (Q1) dla x oraz y
 - wartość drugiego kwartyłu (Q2, mediana) dla x oraz y
 - wartość trzeciego kwartyłu (Q3) dla x oraz y
- komiwojażer zawsze będzie wyruszał z początku układu współrzędnych tj. z miasta o lokalizacji w punkcie (x:0,0; y:0,0)

Do zebrania danych pomiarowych autor zamierza wykorzystać sprzęt o poniżej przedstawionych parametrach.

Laptop: Lenovo Y700-15ISK [5]

CPU: [6]Procesor Intel® Core™ i7-6700HQ (pamięć cache 6 MB, nawet do 3,50 GHz)

Pamięć RAM:

[DIMM1: Kingston HyperX KHX2133C13S4/8G]

Właściwości modułu pamięci:

Nazwa modułu	Kingston HyperX KHX2133C13S4/8G
Numer seryjny	BD020DC7h (3339513117)
Data produkcji	Tydzien 22 / 2018
Rozmiar modułu	8 GB (1 rank, 16 banks)
Typ modułu	SO-DIMM
Typ pamięci	DDR4 SDRAM
Szybkość pamięci	DDR4-2133 (1066 MHz)
Szerokość modułu	64 bit
Napięcie modułu	1,2 V
Metoda detekcji błędów	Brak
Producent DRAM	SK hynix
SDRAM Die Count	1

Taktowanie pamięci:

① 1066 MHz	16-13-13-35 (CL-RCD-RP-RAS) / 50-374-278-171-6-4-6-23 (RC-RFC1-RFC2-RFC4-RRDL-RRDS-CCDL-FAW)
② 1066 MHz	15-13-13-35 (CL-RCD-RP-RAS) / 50-374-278-171-6-4-6-23 (RC-RFC1-RFC2-RFC4-RRDL-RRDS-CCDL-FAW)
③ 1066 MHz	14-13-13-35 (CL-RCD-RP-RAS) / 50-374-278-171-6-4-6-23 (RC-RFC1-RFC2-RFC4-RRDL-RRDS-CCDL-FAW)
④ 1066 MHz	13-13-13-35 (CL-RCD-RP-RAS) / 50-374-278-171-6-4-6-23 (RC-RFC1-RFC2-RFC4-RRDL-RRDS-CCDL-FAW)
⑤ 985 MHz	12-12-12-33 (CL-RCD-RP-RAS) / 46-345-257-158-6-4-6-21 (RC-RFC1-RFC2-RFC4-RRDL-RRDS-CCDL-FAW)
⑥ 903 MHz	11-11-11-30 (CL-RCD-RP-RAS) / 42-317-235-145-5-4-19 (RC-RFC1-RFC2-RFC4-RRDL-RRDS-CCDL-FAW)
⑦ 820 MHz	9-9-9-25 (CL-RCD-RP-RAS) / 35-259-193-119-4-3-16 (RC-RFC1-RFC2-RFC4-RRDL-RRDS-CCDL-FAW)
⑧ 738 MHz	9-9-9-25 (CL-RCD-RP-RAS) / 35-259-193-119-4-3-16 (RC-RFC1-RFC2-RFC4-RRDL-RRDS-CCDL-FAW)

Extreme Memory Profile v2.0:

Nazwa profilu	Enthusiast (Certified)
Szybkość pamięci	DDR4-2133 (1066 MHz)
Wartość napięcia	1,20 V
Zaleczana liczba modułów DIMM na kanał	2
① 1066 MHz	16-13-13-35 (CL-RCD-RP-RAS) / 50-374-278-171-6-4-23 (RC-RFC1-RFC2-RFC4-RRDL-RRDS-CCDL-FAW)
② 1066 MHz	15-13-13-35 (CL-RCD-RP-RAS) / 50-374-278-171-6-4-23 (RC-RFC1-RFC2-RFC4-RRDL-RRDS-CCDL-FAW)
③ 1066 MHz	14-13-13-35 (CL-RCD-RP-RAS) / 50-374-278-171-6-4-23 (RC-RFC1-RFC2-RFC4-RRDL-RRDS-CCDL-FAW)
④ 1066 MHz	13-13-13-35 (CL-RCD-RP-RAS) / 50-374-278-171-6-4-23 (RC-RFC1-RFC2-RFC4-RRDL-RRDS-CCDL-FAW)
⑤ 985 MHz	12-12-12-33 (CL-RCD-RP-RAS) / 46-345-257-158-6-4-21 (RC-RFC1-RFC2-RFC4-RRDL-RRDS-CCDL-FAW)
⑥ 903 MHz	11-11-11-30 (CL-RCD-RP-RAS) / 42-317-235-145-5-4-19 (RC-RFC1-RFC2-RFC4-RRDL-RRDS-CCDL-FAW)
⑦ 820 MHz	10-10-10-27 (CL-RCD-RP-RAS) / 39-288-214-132-5-4-18 (RC-RFC1-RFC2-RFC4-RRDL-RRDS-CCDL-FAW)
⑧ 738 MHz	9-9-9-25 (CL-RCD-RP-RAS) / 35-259-193-119-4-3-16 (RC-RFC1-RFC2-RFC4-RRDL-RRDS-CCDL-FAW)

Rysunek 2 Fragment raportu wygenerowanego z programu Aida64 Extream o zainstalowanej kości pamięci RAM w slocie nr. 1

[DIMM3: Kingston HyperX KHX2133C13S4/8G]

Właściwości modułu pamięci:

Nazwa modułu	Kingston HyperX KHX2133C13S4/8G
Numer seryjny	BE020DC7h (3339518654)
Data produkcji	Tydzien 22 / 2018
Rozmiar modułu	8 GB (1 rank, 16 banks)
Typ modułu	SO-DIMM
Typ pamięci	DDR4 SDRAM
Szybkość pamięci	DDR4-2133 (1066 MHz)
Szerokość modułu	64 bit
Napięcie modułu	1,2 V
Metoda detekcji błędów	Brak
Producent DRAM	SK hynix
SDRAM Die Count	1

Taktowanie pamięci:

① 1066 MHz	16-13-13-35 (CL-RCD-RP-RAS) / 50-374-278-171-6-4-6-23 (RC-RFC1-RFC2-RFC4-RRDL-RRDS-CCDL-FAW)
② 1066 MHz	15-13-13-35 (CL-RCD-RP-RAS) / 50-374-278-171-6-4-6-23 (RC-RFC1-RFC2-RFC4-RRDL-RRDS-CCDL-FAW)
③ 1066 MHz	14-13-13-35 (CL-RCD-RP-RAS) / 50-374-278-171-6-4-6-23 (RC-RFC1-RFC2-RFC4-RRDL-RRDS-CCDL-FAW)
④ 1066 MHz	13-13-13-35 (CL-RCD-RP-RAS) / 50-374-278-171-6-4-6-23 (RC-RFC1-RFC2-RFC4-RRDL-RRDS-CCDL-FAW)
⑤ 985 MHz	12-12-12-33 (CL-RCD-RP-RAS) / 46-345-257-158-6-4-21 (RC-RFC1-RFC2-RFC4-RRDL-RRDS-CCDL-FAW)
⑥ 903 MHz	11-11-11-30 (CL-RCD-RP-RAS) / 42-317-235-145-5-4-19 (RC-RFC1-RFC2-RFC4-RRDL-RRDS-CCDL-FAW)
⑦ 820 MHz	10-10-10-27 (CL-RCD-RP-RAS) / 39-288-214-132-5-4-18 (RC-RFC1-RFC2-RFC4-RRDL-RRDS-CCDL-FAW)
⑧ 738 MHz	9-9-9-25 (CL-RCD-RP-RAS) / 35-259-193-119-4-3-16 (RC-RFC1-RFC2-RFC4-RRDL-RRDS-CCDL-FAW)

Extreme Memory Profile v2.0:

Nazwa profilu	Enthusiast (Certified)
Szybkość pamięci	DDR4-2133 (1066 MHz)
Wartość napięcia	1,20 V
Zaleczana liczba modułów DIMM na kanał	2
① 1066 MHz	16-13-13-35 (CL-RCD-RP-RAS) / 50-374-278-171-6-4-23 (RC-RFC1-RFC2-RFC4-RRDL-RRDS-CCDL-FAW)
② 1066 MHz	15-13-13-35 (CL-RCD-RP-RAS) / 50-374-278-171-6-4-23 (RC-RFC1-RFC2-RFC4-RRDL-RRDS-CCDL-FAW)
③ 1066 MHz	14-13-13-35 (CL-RCD-RP-RAS) / 50-374-278-171-6-4-23 (RC-RFC1-RFC2-RFC4-RRDL-RRDS-CCDL-FAW)
④ 1066 MHz	13-13-13-35 (CL-RCD-RP-RAS) / 50-374-278-171-6-4-23 (RC-RFC1-RFC2-RFC4-RRDL-RRDS-CCDL-FAW)
⑤ 985 MHz	12-12-12-33 (CL-RCD-RP-RAS) / 46-345-257-158-6-4-21 (RC-RFC1-RFC2-RFC4-RRDL-RRDS-CCDL-FAW)
⑥ 903 MHz	11-11-11-30 (CL-RCD-RP-RAS) / 42-317-235-145-5-4-19 (RC-RFC1-RFC2-RFC4-RRDL-RRDS-CCDL-FAW)
⑦ 820 MHz	10-10-10-27 (CL-RCD-RP-RAS) / 39-288-214-132-5-4-18 (RC-RFC1-RFC2-RFC4-RRDL-RRDS-CCDL-FAW)
⑧ 738 MHz	9-9-9-25 (CL-RCD-RP-RAS) / 35-259-193-119-4-3-16 (RC-RFC1-RFC2-RFC4-RRDL-RRDS-CCDL-FAW)

Rysunek 3 Fragment raportu wygenerowanego z programu Aida64 Extream o zainstalowanej kości pamięci RAM w slocie nr. 2

Pamięć ROM:

- **Samsung SSD 960 EVO 250GB [7]**
- **CT500MX500SSD1 500GB [8]**

5. Plan pracy

Omawiane w tej pracy badania zostały przeprowadzone zgodnie z poniżej przedstawionym planem pracy.

1. Sformułowanie celu i zakresu badań
2. Zapoznanie się z istniejącą już pracami opisującymi podobne zagadnienie
3. Opracowanie i wybranie metryk wykorzystanych do badań
4. Przegląd dostępnych bibliotek oferujących gotowe implementacje badanych algorytmów
5. Przygotowanie implementacji architektury pomiarowej z wykorzystaniem języka Python
6. Przygotowanie danych wejściowych
7. Przeprowadzenie badań i zgromadzenie danych pomiarowych
8. Opracowanie danych pomiarowych

6. Omówienie badanych algorytmów

6.1 Algorytm brutalnej siły

6.1.1 Naiwne podejście siłowe

6.1.2 Algorytm Helda-Karpa

6.2 Algorytm zachłanny

6.2.1 Algorytm greedy search

6.3 Algorytm heurystyczny

6.3.1 Algorytm A*

6.3.2 Algorytm przeszukiwań lokalnych

6.3.3 Algorytm symulowanego wyżarzania

6.3.4 Algorytm genetyczny

6.3.5 Algorytm mrówkowy

6.4 Przegląd wyników badań

6.4.1 Wyniki badania optymalności

6.4.2 Wyniki badania złożoności czasowej

6.4.3 Wyniki badania złożoności pamięciowej

7. Konkluzje

8. Bibliografia

- [1] University of Waterloo 200 University Avenue West Waterloo, Ontario, N2L 3G1, [Online]. Available: <http://www.math.uwaterloo.ca/tsp/data/usa/tours.html>. [Accessed 18 11 2021].
- [2] M. Radom, „Kombinatoryczne aspekty nieklasycznego sekwencjonowania DNA przez hybrydyzację,” dr hab. inż. Piotr Formanowicz, profesor PP, Poznań, 2011.
- [3] Z. L. Ratnesh Kumar, „Optimizing the Operation Sequence of a Chip Placement Machine Using TSP Model,” Departmennt of Electrical Engineering University of Kentucky Lexington , KY 40506-0046, Kentucky, 2003.
- [4] W. Z. A. Katedra Informatyki Stosowanej, „Inteligencja obliczeniowa: Heurystyki i metaheurystyki,” 11 01 2021. [Online]. Available: http://www.pi.zarz.agh.edu.pl/intObl/notes/IntObl_w2.pdf. [Data uzyskania dostępu: 11 01 2022].
- [5] L. T. B. V. S. Z. O. O. W. POLSCE, „Ideapad Y700 (15),” LENOVO TECHNOLOGY B V SP Z O O , 11 01 2022. [Online]. Available: <https://www.lenovo.com/pl/pl/laptops/ideapad/y700-series/Ideapad-Y700-15/p/88IPY700618>. [Data uzyskania dostępu: 11 01 2022].
- [6] I. T. P. S. Z. O. O, „Procesor Intel® Core™ i7-6700HQ (pamięć cache 6 MB, nawet do 3,50 GHz),” INTEL TECHNOLOGY POLAND SP Z O O, 11 01 2022. [Online]. Available: <https://www.intel.pl/content/www/pl/pl/products/sku/88967/intel-core-i76700hq-processor-6m-cache-up-to-3-50-ghz/specifications.html>. [Data uzyskania dostępu: 11 01 2022].
- [7] SAMSUNG ELECTRONICS POLSKA SP Z O O, „Samsung SSD 960 EVO 250GB,” SAMSUNG ELECTRONICS POLSKA SP Z O O, 11 01 2022. [Online]. Available:

<https://www.samsung.com/pl/memory-storage/nvme-ssd/960-evo-nvme-m-2-ssd-250gb-mz-v6e250bw/>. [Data uzyskania dostępu: 11 01 2022].

- [8] CRITICAL SYSTEM SOLUTIONS SP Z O O, „Crucial MX500,” CRITICAL SYSTEM SOLUTIONS SP Z O O, 11 01 2022. [Online]. Available: <https://www.crucial.com/ssd/mx500/ct500mx500ssd1>. [Data uzyskania dostępu: 11 01 2022].
- [9] The software quality company, [Online]. Available: <https://www.tiobe.com/tiobe-index/>. [Data uzyskania dostępu: 19 11 2021].
- [10] K. Olszowy, „Aplikacja znajdująca najkrótszą drogę w supermarketie,” Akademia Górnictwo-Hutnicza im. Stanisława Staszica w Krakowie, Kraków, 2017.
- [11] O, INTEL TECHNOLOGY POLAND SP Z O, „Procesor Intel® Core™ i7-6700HQ (pamięć cache 6 MB, nawet do 3,50 GHz),” INTEL TECHNOLOGY POLAND SP Z O O, 11 01 2022. [Online]. Available: <https://www.intel.pl/content/www/pl/pl/products/sku/88967/intel-core-i76700hq-processor-6m-cache-up-to-3-50-ghz/specifications.html>. [Data uzyskania dostępu: 11 01 2022].